DEVELOPMENT OF SUSTAINABLE E-LEARNING CONTENT WITH THE OPEN SOURCE eLESSON MARKUP LANGUAGE eLML

J. Fisler [a], S. Bleisch [b], M. Niederhuber [c]

[a] Department of Geography, University of Zurich, 8057 Zurich, Switzerland – fisler@geo.unizh.ch
[b] Basel University of Applied Sciences (FHBB), 4132 Muttenz, Switzerland – s.bleisch@fhbb.ch
[c] Forest Engineering Group, ETH Zurich, 8092 Zurich, Switzerland – monika.niederhuber@env.ethz.ch

KEY WORDS: Internet/Web, Education, Learning, Teaching, Open Systems, Content-based, Standardization, Sustainable Structure

ABSTRACT:

eLML, the open source "eLesson Markup Language" has been developed out of the proven XML-based content structure of the Swiss Virtual Campus (SVC) Project GITTA (Geographic Information Technology Training Alliance). This paper presents the eLML structure and how sustainable e-Learning content can be developed, exchanged and employed using eLML.
eLML is based on the didactical concept ECLASS (adapted from Gerson, 2000), standing for entry, clarify, look, act, self-assessment and summary. Every self-contained lesson is subdivided into one or more units that are structured according to the didactical concept. To allow different teaching and learning scenarios most of the structure elements are optional or can be repeated several times. The development of a lesson starts with the definition of learning objectives and the planning of content for each part of the didactical structure. This ensures that didactical aspects are considered foremost, in a field where technical problems can sometimes become too prominent otherwise. Storing the content in an XML-based structure provides many advantages, ranging from the sustainable management of a pool of lessons to different presentations of the content according to varying requirements. Out of a pool of eLML-based lessons, where each lesson also contains a glossary, bibliography and metadata section in addition to the actual content, different courses can be assembled. eLML is not a new learning management system (LMS) but eLML-based e-learning lessons can, for example, be transformed into XHTML and be used in combination with a LMS. eLML-based content can also be imported directly into a modern commercial or open source LMS using the IMS Content Packaging Format (CP).
The paper presents experiences from the development of eLML itself, the design of e-learning content based on this structure, and the use of eLML-based content in conjunction with a LMS. The current use of eLML in several SVC projects ensures that eLML will be enhanced and kept up to date for future e-learning projects.

1. INTRODUCTION

Since eLML is based on the technical framework of the Swiss e-learning project GITTA (Geographic Information Technology Training Alliance), this paper begins with a short overview about the GITTA project.

1.1 GITTA – The history behind eLML

The GITTA project is one of around fifty e-learning projects funded by the Swiss Virtual Campus (SVC, 2001). Between 2001 and 2004 a consortium of 10 interdisciplinary Geographic Information Science and Technology (GIST) teaching institutions, from Swiss universities, federal institutes of technology and universities of applied sciences – together covering around 75% of the Swiss GIST offerings in higher education – built up a modular online course in GIST. The main motivation was to exploit synergies, increase the teaching capacity and improve the quality of GIST courses. Nearly 40 authors, from three language regions, created a pool of 50 lessons, divided into basic and intermediate levels, and 10 case studies that can be used on a modular basis for GIST teaching (Fisler, 2004).

This heterogeneous consortium needed some strict didactical and technical structures to create consistent and similar looking lessons and case studies. After an extensive evaluation of both the lessons and the case studies, a didactical model was chosen using XML as the technical backbone. Using Document Type Definitions (DTDs), XML documents conforming to the didactical model could be created and validated. Thus XML offered the sustainability and interoperability needed for GITTA

that no Learning Management System (LMS) offered in 2001. Choosing the didactical guidelines first and then the technology that could cope with these guidelines made GITTA a problem driven and not a technology driven e-learning project.

The first to discover the potential of GITTA's XML structure was SVC's technical board Edutech. They both financially and philosophically supported GITTA to publish their XML framework and make it available to other SVC projects. Since the GITTA XML structure, based on outdated DTDs and depending on specific server technologies (e.g. Apache Cocoon, 1998), was undocumented and hard to learn for authors, a consolidated, server-independent and documented XML framework based on XML Schema was developed in spring 2004.

1.2 eLML – Yet another markup language?

This new XML framework was called eLML, the "eLesson Markup Language" (Fisler & Bleisch, 2004) and was published under the General Public License (GPL) as an open source project (OSI, 1998) on Sourceforge.net. eLML is, in contrast to the old GITTA structure, free of any GIST-specific terms or technologies and therefore usable for any e-learning project choosing XML as their main base for creating and storing content. Furthermore, eLML as an open source project is free of charge and welcomes new contributors in the development process to enhance the framework.

eLML should be understood as the result of three years experience in working with an XML structure and the wish make this knowledge accessible and help authors creating structured e-learning content. eLML can be compared to other markup languages that were developed to describe e-learning content (e.g. LMML in Süss et al., 2001 or EML in Koper, 2002). The main reason for creating a new markup language was that when the GITTA project started, none of these XML markup languages satisfied its needs.

2. ᴇLML – E-LESSON MARKUP LANGUAGE

2.1 ECLASS – the didactical concept behind eLML

The aim of eLML was to offer the GITTA authors a tool that ensured conformity to certain didactic guidelines. These guidelines were adapted from the ECLASS model developed by Gerson (2000). ECLASS is an acronym for the terms entry, clarify, look, act, self-assessment and summary. Enriching these elements with elements like glossary, bibliography or metadata led to the XML framework that is now called eLML.

As shown in Figure 1, GITTA lessons were organized into different modules that again were part of either the basic or the intermediate level. Both level and module are purely organizational entities with no technical relation within eLML. The eLesson Markup Language uses lessons as the smallest interchangeable entity. A lesson is built up of units (conforming to the ECLASS model) and of additional elements like learning objectives, bibliography, glossary and metadata.
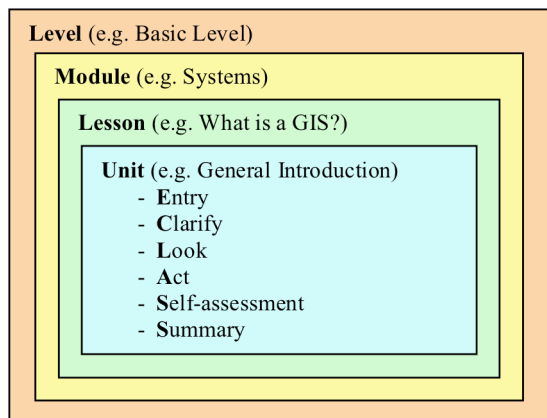

Figure 1: The ECLASS model

eLML is not as rigid as it may look. Since some elements are optional or can be used in reverse order, it is flexible enough to allow the representation of entities such as the following:
- **Lessons**: Standard e-learning lessons begin with an entry element that describes the content of the lesson. Then they could continue with a clarify element describing some theory and one or more look elements to show examples. Usually lessons end with a self-assessment (to check if learning objectives are reached) and a summary.
- **Case studies**: As described in Niederhuber (2005), GITTA case studies use their own didactical model. They usually start with an entry element followed by two units (using clarify elements), whereby the clarify elements are used to describe the instructions.
- **Others**: Even documentations could use eLML to generate structured reports. In this case both act and self-assessment elements would be left out.

2.2 Structure of an eLML lesson

Figure 2 shows how the described didactical model ECLASS is represented within the XML framework eLML:
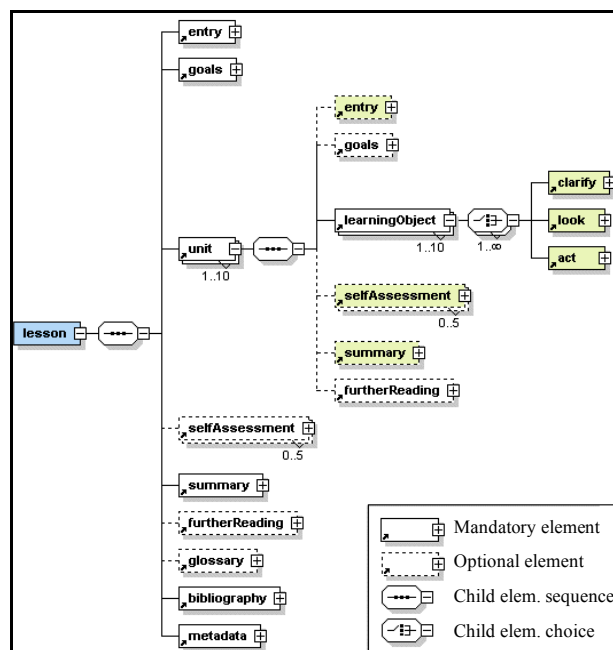

Figure 2: Structure of an eLML lesson

An eLML lesson always starts with the mandatory introduction (entry) followed by a concise listing of the lesson's learning objectives (goals). The unit elements, described in the next paragraph, contain the actual content of a lesson. After the units a lesson can have one or more self-assessments followed by a mandatory summary and an optional further reading section. Also optional is the glossary to describe terms used within the lesson – the XML Schema ensures that all glossary terms used in a lesson are defined in the glossary. The Harvard Citing System (Holland, 2004) is used for the mandatory bibliography that is also validated. All citations, references, further readings etc., have to be listed within the bibliography section, otherwise the XML parser issues an error and thus a lesson will not be valid. Not many authors like to fill out metadata, but eLML ensures that at least the minimal metadata elements are filled out. This subset of the IMS Learning Resource Metadata Specification (2002) can be used to store data about the length of the lessons, the author(s), copyrights, the required knowledge to solve this lesson and the basic technical requirements. Both bibliography and metadata are defined separately and thus can be replaced by other citing references or metadata standards.

With two exceptions, the unit element itself is built up exactly as the lesson root element described above:
1. The glossary, bibliography and metadata elements are only available at the lesson and not the unit level.
2. At the unit level all elements, except the "learning object" (described below), are optional.

Within units, eLML defines so-called "learning objects" (not to be confused with the learning objectives – called "goals" within eLML). Each learning object describes a certain model, concept, term or process using the three elements clarify (theory), look (example) and act (the student has to "do" something). These three elements have no visual representation and are solely used as guiding elements for authors. This way, the author has to think about how a certain concept should be

presented to the student. Whether a learning object starts with some theory (clarify element) and continues with one or more examples (look elements) or, alternatively, the student first has to do something (act element) and then reads the theory behind it (clarify element) is left to the author.

A learning object should fit on about one screen page and take the student about 5 minutes to understand. Units are typically made of 4 to 6 learning objects, typically requiring about 30 minutes of working time. In the case of GITTA, lessons were built of 4 to 6 units, meaning that GITTA lessons take the students normally around two to three hours to work through. But this is just one approach since the total required working time for a lesson is not defined within eLML. Other projects use eLML lessons to represent a whole semester course, thus one unit represents one lecture (one or two hour).

2.3 Content elements

The last section covered the basic structure of an eLML lesson. The afore mentioned structural elements, entry, unit, learning object, self-assessment, summary etc., could be represented as lesson chapter titles. Within these chapters, there are content elements that contain the actual subject matter.

The old GITTA structure worked with semantic elements using e.g. explanation, remark or motivation paragraphs. This approach was rejected because for the authors it was too complicated to grasp and because the elements did not have a visual representation (or if they had, the authors chose the elements according to their representation in the final layout and not according to their semantic meaning). In theory a total separation between content and representation (layout) would be desirable but – as the GITTA project showed within its three years – this is not realistic. Therefore certain structural elements (column, formatted, newLine etc.) are offered within eLML but their use is not recommended.

A short overview of the eLML content elements:
- column: Defines a two- or three-column layout.
- table: Used for tables and not as a layout element.
- list: Numbered or bulleted lists.
- box: Content of this element is represented in a box. The exact layout of boxes is defined separately.
- term: When a glossary term is used the definition is either available as "mouse over" link or as a separate paragraph with a link to the glossary.
- newLine: A line break (not recommended).
- multimedia: Pictures, Flashes, Applets, Movies or even plain HTML code (e.g. JavaScript).
- formatted: Possibility to format text as bold, italic, underlined, subscript etc. Not recommended.
- popup: Clicking on the title (e.g. a question) opens a box (e.g. the answer). Always open in print version!
- link: Link to external or internal resources including other units, learning objects etc.
- citation: Citations can be inline or as a paragraph with many options described in the manual. A resource has to be defined in the bibliography before being used.
- paragraph: A regular paragraph that offers attributes to be visible only to tutors or to be shown only in the print or the online version of a lesson.
- indexItem: Marking words to be listed in the index.

All of these elements have additional attributes. For instance, the "icon" attribute allows certain elements to be emphasised with a special icon: "attention" by an exclamation mark icon or "help" by an interrogation mark icon. Furthermore, this attribute allows a list (e.g. in the appendix) of all elements with the same icon attribute to be extracted. A scenario where this might be used is in a math course where the important formulae were marked with the attention icon. The appendix could then list only the important formulae as a summary without text.

eLML also defines rules for nesting of elements. For instance, to include a column within a list element would not make sense, neither would a table within a multimedia element. Therefore the eLML XML Schema exactly defines which element can be used where. An example of how these elements are used and how such an XML document looks can be seen in Figure 3.

```xml
<?xml version="1.0" encoding="UTF-8"?>
<lesson xmlns:xsi="http://www.w3.org/2001/XMLSchema-insta
    title="What is a GIS?">
    <entry label="LEntryLesson" title="Introduction">
        <paragraph> Living in what we call 'the age of in
            storage capacities, high resolution sensors o
            to be acquired, stored, organised and structu
            only to manage data, but also for the interpr
        <paragraph icon="remark" role="student">In inform
            style="italic">unprocessed, uninterpreted
            >information</formatted> can be extracted. Ho
            source, depending on factors like perspective
        <paragraph>Thus, an information system is a tool
            is usually capable of integrating and combini
            transformed into useful information by a user
            a whole suite of software packages and hardwa
        <paragraph>
            <multimedia mimeType="gif" src="../image/less
        </paragraph>
```

Figure 3: An extract of an eLML XML document.

2.4 Developing of an eLML lesson

Creating an eLML lesson typically starts with defining the learning objectives of a particular lesson (the "goals" element within eLML). When the author has defined what the student should learn within this lesson, he/she decides on the units that are needed to present the subject matter and from there each learning object that presents a certain aspect of the topic. The actual writing of the lesson is done with an XML editor. Multimedia elements like movies or flash animations etc. are made using appropriate tools.

For authoring a lesson's XML document, common editors that support XML Schema validation can be used. This implies that an author has a basic knowledge of XML. A web front end, as used for other e-learning projects, could not be realised due to the complexity of the XML Schema. Another solution is to use tools that can render a Graphical User Interface (GUI) based on an XML Schema. JAXFront (2001), see Figure 4, is such a tool. It was tested with eLML and might be used by GITTA and other project authors in future.
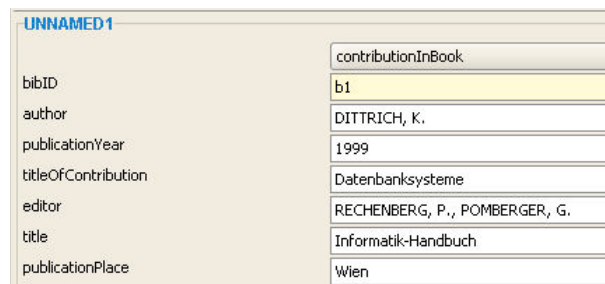
UNNAMED1	
	contributionInBook
bibID	b1
author	DITTRICH, K.
publicationYear	1999
titleOfContribution	Datenbanksysteme
editor	RECHENBERG, P., POMBERGER, G.
title	Informatik-Handbuch
publicationPlace	Wien

Figure 4: JAXFront Screenshot of eLML bibliography reference

2.5 Presentation and layout

eLML includes XSLT and XSL-FO files for transformation into both an online (XHTML) and a print (PDF) version of a lesson. The standards used are described in detail in Section 3.3.

Online version: The online version uses templates to generate the final XHTML version of a lesson. eLML includes a plain layout, two GITTA layouts (Figure 5) and another projects template. Three means of creating a new template are provided:

a) CSS: Using the plain layout and adapting the included CSS file is the simplest way to create a customised layout.

b) XHTML: Creating an XHTML template using a web design tool is also fairly easy. Adding two XSL statements transforms it into an eLML template.

c) XSLT: Using XSLT statements it is possible to build a new template with new functions or elements.

Figure 5:   GITTA lesson transformed using the GITTA layout

Thanks to the new XSLT 2.0 standard, eLML since 2005 offers a multiple output document option. The page breaks can either be at the lesson level (one single page generated), at the unit level (each unit on a single page) or at the learning objects level (each learning object on a single page). Also since 2005, eLML output can be zipped into an IMS Content Package and imported into any existing LMS, such as WebCT or OLAT (1999; see Figure 6).

Figure 6:   Same lesson as shown in Figure 5 imported into the LMS OLAT using IMS Content Package standard.

Print version: The print version uses the Apache Formatting Object Processor (Apache FOP, 2001) to generate a PDF document, like the one shown in Figure 7. This engine does not support all the latest XSL-FO elements, as can be seen on the Apache FOP compliance page. Nevertheless we chose to use it because:

- Open source products should be based on other open source products and not require commercial software.
- The Apache FOP is the most widely used engine.
- All elements used in eLML were supported by Apaches FOP either directly or using a workaround.
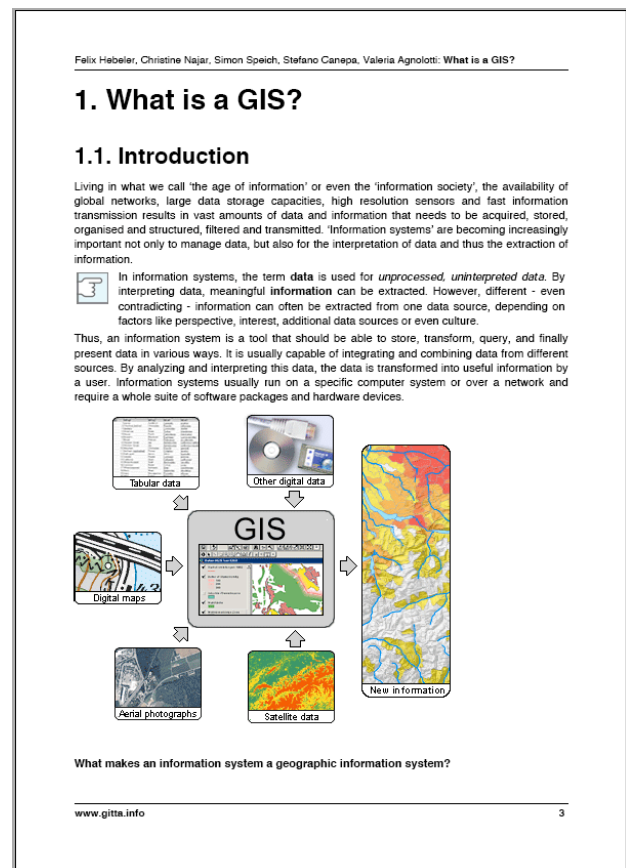

Figure 7:   Again the same GITTA lesson, this time as a PDF using the Apache Formatting Object Processor.

There are additional parameters that can be adjusted, having impact on the online or print version. These include a specific contact email address used in contact links, disabling the chapter numbering, language settings, project URL for home links, defining value lists, etc. These options are described in detail in the eLML manual (Bleisch & Fisler, 2004).

3.   TOWARDS AN OPEN SOLUTION

When the GITTA project started in 2001 nobody talked about open source or open content. The consortium agreed to produce content for internal use only, like most SVC projects at that time. As funding ended, a solution for sustainable low cost maintenance had to be found. That is when open source and open content came into play.

3.1 Releasing eLML under an open source license

Keeping the GITTA technical infrastructure up to date required a lot of resources. When Edutech came up with the idea of opening the XML framework to others, the consortium agreed that open source might be the key for sustainable GITTA maintenance. Thus the technical "backbone" eLML was released under the open source license GPL on Sourceforge.net (see Figure 8). If it could be possible to motivate other e-learning projects to use eLML and to participate in developing and updating eLML, the maintenance cost for GITTA would be lowered. By opening eLML for the e-learning community via open source, GITTA saved other projects a lot of time required to building up a technical infrastructure. This time can be invested by these projects into the enhancement of eLML and keeping it compatible to the latest e-learning standards. GITTA benefits from these developments since there is no need to finance its own technical development team.

Figure 8: The Sourceforge.net eLML project page

3.2 Releasing content under the Creative Commons license

The open source philosophy should not only cover the technical part of an e-learning project but could also be applied to its content. Benefits of sharing knowledge and content include the opportunity of improved content due to the feedback from external authors. Therefore, in April 2005 the GITTA consortium decided to release all its material under the Creative Commons license (CC, 2001). As the open source licenses mainly address software projects, the Creative Commons license was developed for publishing music, movies, photos, and other content such as electronic books, educational material and e-learning courses. Via the CC website it is possible to assemble a personal license that either includes or excludes commercial use, altering of content, feeding back updates to the project, etc. CC offers the whole spectrum of possibilities from full copyright (all rights reserved) to public use. The license founded in 2001 seems to have become a standard not only in the USA and Europe but also throughout the world (e.g. in Brazil as described in Ermert, 2004).

3.3 Using common standards

eLML uses common (open source) standards wherever possible. Each lesson is stored in a single XML file, conforming to the eLML XML Schema. This file is then transformed either using XSL-FO into PDF or using XSLT 2.0 in one or more XHTML output documents. Also exported is the imsmanifest.xml file used for the IMS Content Package format. The versions and references to these standards are shown in Figure 9.
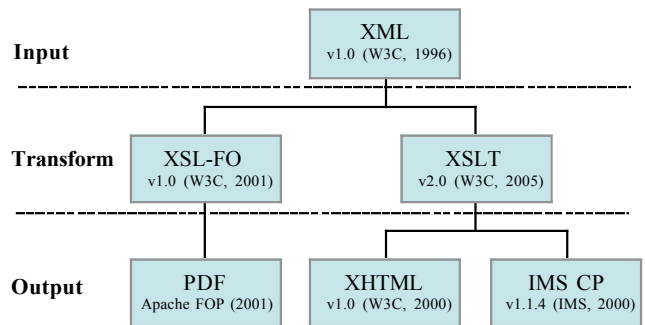

Figure 9: Standards used by eLML

4. CONCLUSIONS

After an introduction about the origins and motivation behind the eLesson Markup Language (eLML) this paper presented eLML in detail. The didactical model behind eLML is an extension of an existing model. This has to be kept in mind when using eLML. Loosening this tight eLML structure is one of the most common requests by new users. While this has already been done to a certain degree, too many changes could dilute the structure in such a way that the didactical concept behind would no longer be visible and stringent.

An eLML lesson is stored as an XML document that has to be validated by an XML Schema. This technique assures that all lessons are built up in a common way. Thus to create a lesson, an author has to be acquainted with the basics of XML and use an appropriate editor to create a lesson. The lack of tools with simple front-ends (GUIs) is one of the most severe drawbacks of eLML at the moment. A Java GUI is presented in this paper, but it is not yet fully tested and would still not offer the same ease-of-use as a standard word processing or web design tool.

The transformation into both an online and a print layout is based on common standards and is accomplished by using open source processors. The benefits are the low costs and the high compatibility with browsers and other tools. Drawbacks of open source can be that updates of tools (like Apache FOP) sometimes require more time to be accomplished in comparison to commercial tools.

Finally, the reasons for releasing eLML under the open source license GPL were discussed. The motivation for this step being that a growing community of eLML users may reduce maintenance costs for a project. As the list of developers started growing in the months after the release of eLML on Sourceforge.net, the benefits were obvious: more updates due to a larger community finding (and repairing) inconsistencies and bugs and a growing list of new implemented features. For example the implementation of the IMS Content Package standard, a major benefit for all projects using eLML, is a feature that was implemented with the help of such a new developer.

The GITTA consortium decided that also the content itself should be released as "open content" under the Creative Commons license. Before doing so, all copyright issues have to be resolved and the lessons have to be updated according to the latest eLML release. The CC license can request that changes to a source have to be sent back to the authors. Thus the hope is that by building up a GITTA user community it will be possible

to gain external authors to feed back corrections and updates of existing lessons or even enlarge the pool of GITTA lessons. Obviously the injection of errors into existing lessons (deliberately or not) could become a problem. Possible solutions would be to release strict rules for gaining write access and/or to set up a review committee that has to approve updates.

To stay informed about eLML, visit www.elml.ch or subscribe to the eLML Newsletter by sending and email with the subject "subscribe" to ELML-news-request@lists.sourceforge.net.

5. REFERENCES

All URLs listed below were last accessed on April 21st 2005!

Apache Cocoon, 1998. The Apache Cocoon project website http://cocoon.apache.org/

Apache FOP, 2001. Formatting Object Processor Version 0.20.5 http://xml.apache.org/fop/

Bleisch, Susanne, Fisler, Joël, 2004. Documentation - eLesson Markup Language eLML. http://sourceforge.net/projects/elml/

Creative Commons License, 2001. http://creativecommons.org

Ermert, Monika, 2004. Geisel einer proprietären Lösung. *c't Magazin* 2/2004 (p. 44), Heise Verlag, Hannover. http://www.heise.de/ct/04/02/044/

Fisler, Joël, 2004. The GITTA website. Geographisches Institut der Universität Zürich. http://www.gitta.info

Fisler, Joël, Bleisch, Susanne, 2004. The eLML website. http://www.elml.ch

Gerson, S. M., 2000. ECLASS: Creating a Guide to Online Course Development For Distance Learning Faculty. *Online Journal of Distance Learning Administration*, Vol. III, No. IV (Winter 2000). State University of Western Georgia. http://www.westga.edu/~distance/ojdla/winter34/gerson34.html

Holland, Matt, 2004. Citing References - The Harvard System. Bournemouth University. http://www.bournemouth.ac.uk/library/using/harvard_system.html

IMS, 2000. Content Packaging Specification. Global Learning Consortium. http://www.imsglobal.org/content/packaging/

IMS, 2002. Learning Resource Metadata Specification. Global Learning Consortium. http://www.imsglobal.org/metadata/

JAXFront, 2001. JAXFront - an XML Rendering Technology. Xcentric GmbH, Zürich, Switzerland. http://www.jaxfront.com/

Koper, Rob, 2002. Educational Modelling Language (EML): adding instructional design to existing specifications. Open University of the Netherlands. http://eml.ou.nl

Niederhuber, M., Heinimann, H.R., Hebel, B., 2005. e-Learning basierte Fallstudien zur akademischen Ausbildung in der Geoinformatik: Methodisches Konzept, Umsetzung und Erfahrungen. Submitted to the 3. Deutsche e-Learning Fachtagung Informatik, 13.-16. September 2005 in Rostock.

OLAT, 1999. Open Source LMS OLAT - Online Learning And Training. Multimedia & E-Learning Services (MELS) der Universität Zürich. http://www.olat.org/

OSI, 1998. Open Source Initiative. http://www.opensource.org/

Süss, Christian, Freitag, Burkhard 2001. Learning Material Markup Language LMML. *IFIS Report* 2001/03. Institut für Informationssysteme und Softwaretechnik (IFIS), Universität Passau. http://www.lmml.de/

SVC, 2001. GITTA project outline. Swiss Virtual Campus, http://www.virtualcampus.ch/display.php?pname=200128pres

W3C, 1996. Extensible Markup Language (XML) 1.0. World Wide Web Consortium. http://www.w3.org/TR/2004/REC-xml-20040204/

W3C, 2000. XHTML 1.0 The Extensible HyperText Markup Language (Second Edition). World Wide Web Consortium. http://www.w3.org/TR/xhtml1/

W3C, 2001. Extensible Stylesheet Language (XSL). World Wide Web Consortium. http://www.w3.org/TR/xsl/

W3C, 2005. XSL Transformations (XSLT) Version 2.0. World Wide Web Consortium. http://www.w3.org/TR/xslt20/

6. ACKNOWLEDGEMENTS